Combining Pretrained and Graph Models for Text Classification

Kaifeng Hao(✉), Jianfeng Li, Cuiqin Hou, Xuexuan Wang, and Pengyu Li

Ping An Technology (Shenzhen) Co., Ltd., Shenzhen, China
{HAOKAIFENG551,LIJIANFENG777,HOUCUIQIN042,WANGXUEXUAN445,
LIPENGYU448}@piangan.com.cn

Abstract. Large-scale pretrained models have led to a series of breakthroughs in Text classification. However, Lack of global structure information limits the performance of pertrained models. In this paper, we propose a novel network named BertCA, which employs Bert, Graph Convolutional Networks (GCN) and Graph Attention Networks (GAT) to handle the task of text classification simultaneously. It aims to learn a rich sentence representation involved semantic representation, global structure information and neighborhood nodes features. In this way, we are able to leverage the complementary strengths of pretrained models and graph models. Experimental results on R8, R52, Ohsumed and MR benchmark datasets show that our model obtains significant performance improvement and achieves the state-of-the-art results in four benchmark datasets.

Keywords: Pertrained models · Graph models · Semantic representation · Global structure information

1 Introduction

Text classification is a basic task in natural language processing (NLP). Multiple deep learning models have been applied to text classification tasks, such as Convolutional Neural Networks [1] (CNN); Recurrent Neural Networks [2] (RNN) and Long Short-Term Memory [3] (LSTM). Recently, the pretrained models (e.g., Bert, GPT-2 and GPT-3) have led to a series of breakthroughs in NLP tasks and obtain state-of-the-art (SOTA) results. Although the pretrained model can obtain contextual sentence representation, it could not process the long text input well and lack global structure information. To addressing this problem, we introduce the Graph Neural Networks (GNN) in this paper.

Recently, GNN has attracted widespread attention. It is effective in NLP tasks which require massive relations and can preserve global structure information in graph embeddings. Graph Convolutional Networks [4] (GCN) can capture high order structure information by combining GNN and CNN. Graph Attention Networks [5] (GAT) introduce the attention mechanism to compute the

Supported by Ping An Technology (Shenzhen) Co., Ltd.

T. Mantoro et al. (Eds.): ICONIP 2021, CCIS 1516, pp. 422–429, 2021.
https://doi.org/10.1007/978-3-030-92307-5_49

hidden representations of each node in the graph by attending over its neighborhood. Thence GCN and GAT can enhance the structure information in different dimensions. However, GCN-style models (such as TextGCN [6]) use one-hot representation to initialize word and document nodes features, This manner will make node features lack semantic level information. Lin [7] proposes BertGCN to solve this problem. This network uses the hidden layer embeddings of Bert [8] as initial nodes features, However, as the increasing of hidden layers, there is still a problem of over-smooth. In GAT model, neighborhood nodes can enhance the center node embedding, This will increase the divergence between nodes and non-adjacent nodes and address the over-smooth. Therefore, we employ Bert, GCN and GAT to handle the task of text classification simultaneously. In this way, we are able to leverage the complementary strengths of pretrained models and graph models.

In this paper, we propose a novel network named BertCA, which employs GCN to learn global structure information based on the hidden layer embeddings of Bert, and computes the hidden representation of each node through GAT for avoiding over-smooth. The result of GAT is treated as a significant weight contained structure information, which is combined with [CLS] embeddings for the final decision. Our work is summarized as follows:

- We propose BertCA, a novel model which combines the powers of pre-trained models and graph networks for text classification.
- The experimental results show that BertCA achieves the state-of-the-art results in several text classification tasks.

2 Related Works

Pretrained Models. Recently researchers have discovered the advantages of combining pretrained models (PTMs) learned on large-scale datasets with downstream models for text classification tasks. Early PTMs focused on learning context-free word embeddings, such as GloVe [9], which aims to obtain global vectors for word representation, GloVe has push lots of models to SOTA on similarity tasks and named entity recognition. Then ELMo [10], which pretrain on a large text corpus and learn functions of the internal states of a deep bidirectional language model. ELMo significantly improve the state of the art across six challenging NLP problems and take a significant step toward context-aware word embeddings.

With the emergence of the Transformer [11], GPT [12] and Bert have brought text classification tasks into a new era. These models focus on modifying the Transformer decoder and encoder, respectively. Later, XLNet [13] learns contextual feature by maximizing the expected likelihood over all permutations of the factorization order and employs transformer-XL to overcome the length limitations of BERT. RoBerta [14] finds that Bert is significantly undertrained and robustly optimizes the training procedure of Bert based on random mask and massive amount corpus. ALBert [15] presents factorized embedding parameterization and cross-layer parameter sharing for reducing the number of parameters

and increasing the training speed of Bert. In a word, powerful pretrained models have greatly promoted the development of NLP.

Graph Models. Models mentioned above already have outstanding performance in processing text classification tasks. However, these models lack ability of learning global structure information. GCN can capture the relationship between graph nodes, this structured graph networks also provide a new perspective for others NLP tasks.

TextGCN is a successful example, which addresses the text classification problem by learning the document-word relationship in the text graph based on word co-occurrence. However, the word and document nodes in graph are initialized with straightforward manner like one-hot representations. Different with TextGCN, The nodes of BertGCN are initialized with the output vector of Bert hidden layer. It combines the advantages of both PTMs and GCN, and achieve SOTA results in this manner. Although several GCN models give outstanding performance, the model has unnecessary complexity and redundant computation. SGC [16] reduces the complexity by converting the nonlinear into linear transformation which not only matches GCN in performance, but it is also faster. Our work is inspired by the work of BertGCN, unlike BertGCN, we employ both GCN and GAT models in the same network.

3 Our Approach

We show the network framework in Fig. 1. Our network employ Bert-style model (e.g., Bert, RoBerta) to initialize the nodes features in text graph, which are used as input of GCN. Then the output of GCN is treated as input for GAT, and the document representations will be iteratively updated based on GCN and GAT, respectively. The outputs of GAT will be sent to softmax function and make a hadamard product with the [CLS] feature of Bert-style models. Finally we add this feature with initial [CLS] feature like Resnet [17], and send the final sentence representation to classifier for predictions. In this manner, we obtain a sentence representation with semantic-level and global structure information which content high order neighborhood nodes information.

3.1 Bulid Graph

We construct a text graph containing word and document nodes following TextGCN. We define word-document edges by the term frequency-inverse document frequency (TF-IDF), and construct word-word edges based on positive point-wise mutual information (PPMI). The weight of an edge between two nodes i and j is defined as:

$$A_{i,j} = TextGCN(i, j) \tag{1}$$

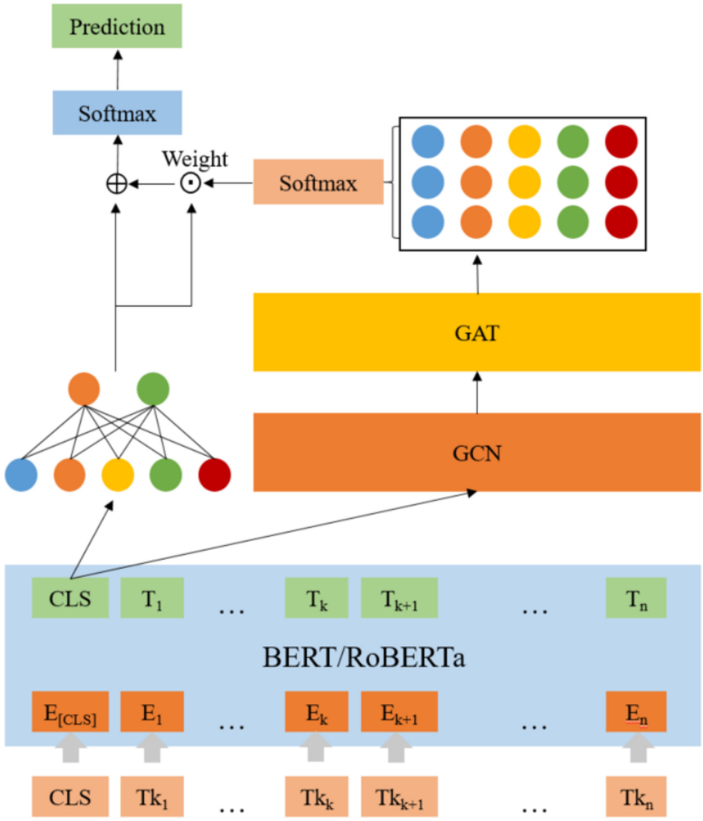

Fig. 1. The framework of BertCA network.

3.2 Bert Layer

We first apply the Bert model to convert the input sentence to word-level embeddings and contextual representation. We obtain the final hidden states h_t from the input sequence of N tokens w_t, and the first [CLS] token is sent to multi-layer perceptron (MLP) for getting the processed feature f_{cls}:

$$h_{cls}, h_i = BERT(w_i) \tag{2}$$

$$f_{cls} = MLP(h_{cls}) \tag{3}$$

3.3 GCN Layer

We replace the node feature with the f_{cls} and feed it into GCN model. The output feature matrix of the i-th GCN layer L^1 is computed as:

$$L^i = \sigma(\overline{A}L^{i-1}W^i) \tag{4}$$

Where σ is the activation function, $\overline{A}$ is the normalized adjacency matrix and W^i is the weight matrix. $L^0 = f_{cls}$ is the initial input of the graph network and we utilize one layer GCN in our network.

3.4 GAT Layer

We feed the output of GCN layer as the input of GAT model. The output feature matrix is updated as:

$$\alpha_{i,j} = \frac{exp(LeakyReLU(\alpha^T W h_i || W h_j))}{\sum_{k \in N_i} exp(LeakyReLU(\alpha^T W h_i || W h_j))} \tag{5}$$

$$h_i^{'} = \sigma(\frac{1}{K} \sum_{k=1}^{K} \sum_{j \in N_i} \alpha_{i,j}^k W^k h_j) \tag{6}$$

Where α is the parameter matrix, σ is the activation function, K is the number of multi-head attention, N_i is the adjacent node of node i and W is the weight matrix. We apply one layer GAT in our network.

3.5 Output Layer

We employ softmax function to normalize the output of GAT. Then we make a hadamard product with the f_{cls} and the normalize matrix. Finally we add this feature with f_{cls} to get the final decision:

$$W_g = softmax(GAT(GCN(f_{cls}, A))) \tag{7}$$

$$R = f_{cls} * W_g + f_{cls} \tag{8}$$

4 Experiments

4.1 Dataset

Our experiments employ four benchmark datasets: R8, R52, Ohsumed and Movie Review (MR). The scale and metrics of datasets are detailed in Table 1:

R8. It is a text classification data set containing 8 kinds of labels. Which source is the finance news and it is part of the benchmark dataset ApteMod.

R52. It is a text classification data set containing 52 kinds of labels. Which source is the finance news and it is other part of the benchmark dataset Apte-Mod.

Ohsumed. The OHSUMED dataset contains the titles and abstracts of 270 medical journals during the five years from 1987 to 1991. Which consists of 8 fields.

MR. It is a movie review classification data set containing two types of labels, and the difference between these movie reviews is obvious.

Table 1. An example of three line table

Dataset	Docs	Training	Test	Words	Nodes	Classes	Average length
R8	7674	5485	2189	7688	15362	8	66
R52	9100	6532	2568	8892	17992	52	70
Ohsumed	7400	3357	4043	14157	21557	23	136
MR	10662	7108	3554	18764	29426	2	20

We use BERT and RoBERTa as our pretrained models, and employ GCN and GAT as the graph models. First, we complete fine-tune stage on dataset based on single pretrained model, and then use it to initialize the Bert parameters in BertCA, finally we train the whole network on the target dataset. The training was conducted on two NVIDIA Tesla V100 GPUs with a batch size of 32. The learning rete of pretrained models is 2e−5 in single model fine-tune stage, the learning rete of pretrained models and graph models is 2e−6 and 1e−3 in training BertCA stage, respectively.

4.2 Results

Table 2. An example of three line table

Model	R8	R52	Ohsumed	MR
TextGCN	97.1	93.6	68.4	76.7
SGC	97.2	94.0	68.5	75.9
Bert	97.8	96.4	70.5	85.7
BertGCN	98.1	96.6	72.8	86.0
BertCA	**98.5**	**97.0**	**73.2**	**87.4**
RoBerta	97.8	96.2	70.7	89.4
RoBertaGCN	98.2	96.1	72.8	89.7
RoBertaCA	**98.3**	**96.7**	**73.0**	**89.9**

The comparison results of TextGCN, SGC, Bert, RoBerta, BertGCN, RoBertaGCN and our model are detailed in Table 2. The results show that our BertCA networks obtain universal performance improvement and achieve SOTA results on text classification benchmark corpus. The main reason is that our network leverages the complementary strengths of Bert, GCN and GAT. Especially, our method has the most obvious improvement in short text corpus like MR. This is because of the additional feature obtained from GAT. Which enhance the center node embedding and prevent over-smooth. On the contrary, the long text corpus like Ohsumed and R52 have slight improvement. That may

because that the long text have adequate information than short text, and the additional feature is not required.

In this way, the [CLS] feature can obtain global structured information from GCN and neighborhood nodes information from GAT, respectively. Therefore, the final sentence representation can successfully satisfy the needs of semantic or structural information in various tasks.

5 Conclusion

In this paper, we propose a novel network named BertCA, which can help learn a rich sentence representation involved semantic representation, global structured information and neighborhood nodes features. Experimental results on four benchmark datasets show that our network obtains significant performance improvements and achieve SOTA results, especially on short text corpus. In the future, we look forward to learning the global structured information and neighborhood features in one model simultaneously, and constructing the weight edges between nodes in a semantic level, and it also worth exploring other short text NLU tasks based on BertCA.

References

1. Krizhevsky, A., Sutskever, I., Hinton, G.: ImageNet classification with deep convolutional neural networks. Adv. Neural Inf. Process. Syst. **25**(2), 1097–1105 (2012)
2. Zaremba, W., Sutskever, I., Vinyals, O.: Recurrent neural network regularization. Eprint Arxiv (2014)
3. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. **9**(8), 1735–1780 (1997)
4. ThoKipf, M.N., Welling, M.: Semisupervised classification with graph convolutional networks. arXiv preprint arXiv:1609.02907 (2016)
5. Velickovic, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., Bengio, Y.: Graph attention networks. arXiv preprint arXiv:1710.10903 (2017)
6. Yao, L., Mao, C., Luo, Y.: Graph convolutional networks for text classification. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 33, pp. 7370–7377 (2019)
7. Lin, Y., Meng, Y., Sun, X., et al.: BertGCN: transductive text classification by combining GCN and BERT. In: Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics (2021)
8. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding (2018)
9. Pennington, J., Socher, R., Manning, C.D.: GloVe: global vectors for word representation. In: EMNLP (2014)
10. Peters, M.E.: Deep contextualized word representations. In: NAACL-HLT (2018)
11. Vaswani, A., et al.: Attention is all you need. In: NeurIPS (2017)
12. Radford, A., Narasimhan, K., Salimans, T., Sutskever, I.: Improving language understanding by generative pre-training (2018)
13. Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R., Le, Q.V.: Generalized autoregressive pretraining for language understanding. In: NeurIPS, XLNet (2019)

14. Liu, Y., et al.: RoBERTa: A robustly optimized BERT pretraining approach. arXiv preprint arXiv:1907.11692 (2019)
15. Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P., Soricut, R.: ALBERT: a lite BERT for self-supervised learning of language representations (2019)
16. Wu, F., Zhang, T., de Souza, A.H., Jr., Fifty, C., Yu, T., Weinberger, K.Q.: Simplifying graph convolutional networks. arXiv preprint arXiv:1902.07153 (2019)
17. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. IEEE (2016)